Speeding up Serpent

- Serpent structure
- CPU models
- Search method
- Important optimizations
- Results

Serpent round function

- Key mixing
- S-box
- Linear transformation

S-box Performs the equivalent of 32 4 $\rightarrow$ 4bit ($Z_{16} \rightarrow Z_{16}$) table lookups using boolean operations.

Advantages of accurate model

- Precise metric of cost
- No register spills
- Utilizes available parallelism

	Gladman	Osvik/x86
instruction type	3AC	2AC
# registers	7-10	5
parallelism	1	2

3AC: a := b *op* c (RISC)

2AC: a := a *op* b (x86)

S-box example

S_0					
r3	^=	r0	r4	=	r1
r1	&=	r3	r4	^=	r2
r1	^=	r0	r0	|=	r3
r0	^=	r4	r4	^=	r3
r3	^=	r2	r2	|=	r1
r2	^=	r4	r4	=~	r4
r4	|=	r1	r1	^=	r3
r1	^=	r4	r3	|=	r0
r1	^=	r3	**r4**	^=	r3
r1, r4, r2, r0					

Search

- Find optimal instruction sequence implementing Serpent S-box
- Level-order traversal
  (level = number of instructions)
- Remove redundant instruction sequences
- Early cutoff by lookahead
- Heuristic advance requirement

Lookahead

- More general CPU model
- Same operations
- Very efficient
- Increases available search depth

CPU model	Osvik/x86	lookahead
instruction type	2AC	3AC
# registers	5	∞
parallelism	2	∞

Advance

- Assume best sequences find results early
- Require sequences to produce result bits
- Increase requirement with depth
- Balance between wide and narrow

Pro Drops bad sequences

Con May drop best sequences

Implementation	Encryption cycles		
	486	Pentium	PPro
AES submission		1605	1170
Gladman	12900	1279	945
Osvik	1650	907	759
Osvik, asm		800	

Implementation	Key setup cycles/PPro
Gladman	1290
Osvik	954

Serpent on ASIC

- S-box NAND network only 5 deep

Serpent on IA-64

- S-boxes may cost only 3 cycles

Conclusion

- Optimized search for Serpent S-boxes
- Speedup of Serpent

486	700%	
Pentium	60%	(asm)
PPro	24%	

- Speedup of key schedule

PPro	35%

- Best S-boxes depend on architecture